Algorithme.

L’ordinateur est un **outil**,
un **exécutant** pointilleux,
méticuleux,
très rapide,
mais totalement dénué d’imagination et d’initiative.

La machine ne fait rien par elle-même;
elle ne fait qu’**appliquer** à la lettre des ordres simples et précis.

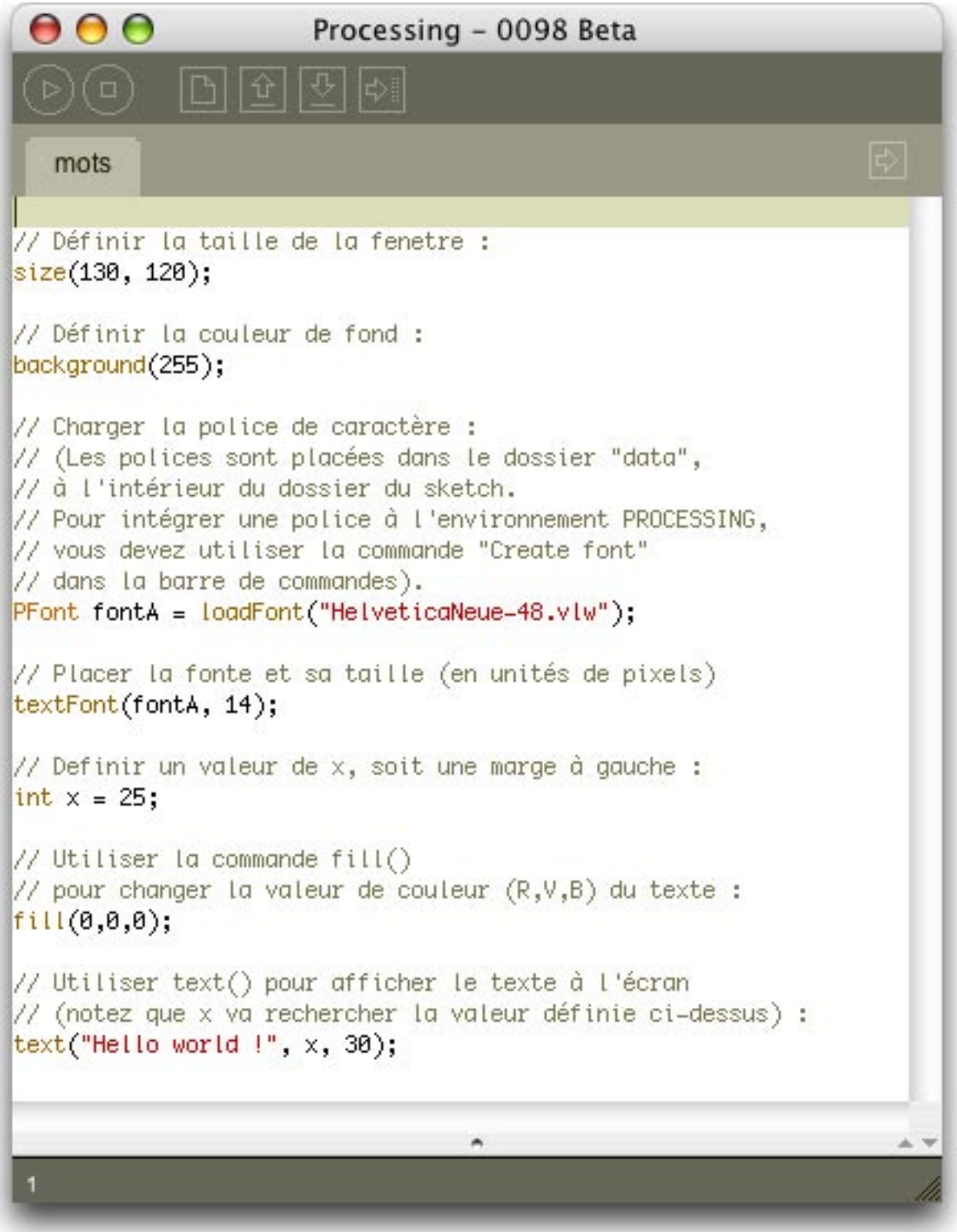
... sauf Hal !

HAL 9000 est le nom du puissant ordinateur doté d'intelligence artificielle, gérant le vaisseau spatial Discovery 1 dans «2001, a space odyssey» de Stanley Kubrick (1968). HAL signifie «Heuristically programmed ALgorithmic computer» et, dans la version française, Carl, soit «Cerveau Analytique de Recherche et de Liaison», ce qui est une assez habile traduction. L'acronyme HAL correspondrait à IBM par décalage d'un rang de chacune des lettres : H->I, A->B, L->M), ce n'est peut-être pas un hasard. Stanley Kubrick et Arthur Charles Clarke ont cependant toujours démenti cette allégation.

http://en.wikipedia.org/wiki/HAL_9000
http://www.robothalloffame.org/hal.html

L’ordinateur reçoit des **instructions** et les **exécute**,
séquentiellement,
sans rechigner
et sans intelligence.

Une suite de telles instructions est un **programme**.


Processing - 0098 Beta
mots
// Définir la taille de la fenetre :
size(130, 120);
// Définir la couleur de fond :
background(255);
// Charger la police de caractère :
// (Les polices sont placées dans le dossier "data",
// à l'intérieur du dossier du sketch.
// Pour intégrer une police à l'environnement PROCESSING,
// vous devez utiliser la commande "Create font"
// dans la barre de commandes).
PFont fontA = loadFont("HelveticaNeue-48.vlw");
// Placer la fonte et sa taille (en unités de pixels)
textFont(fontA, 14);
// Definir un valeur de x, soit une marge à gauche :
int x = 25;
// Utiliser la commande fill()
// pour changer la valeur de couleur (R,V,B) du texte :
fill(0,0,0);
// Utiliser text() pour afficher le texte à l'écran
// (notez que x va rechercher la valeur définie ci-dessus) :
text("Hello world !", x, 30);
1

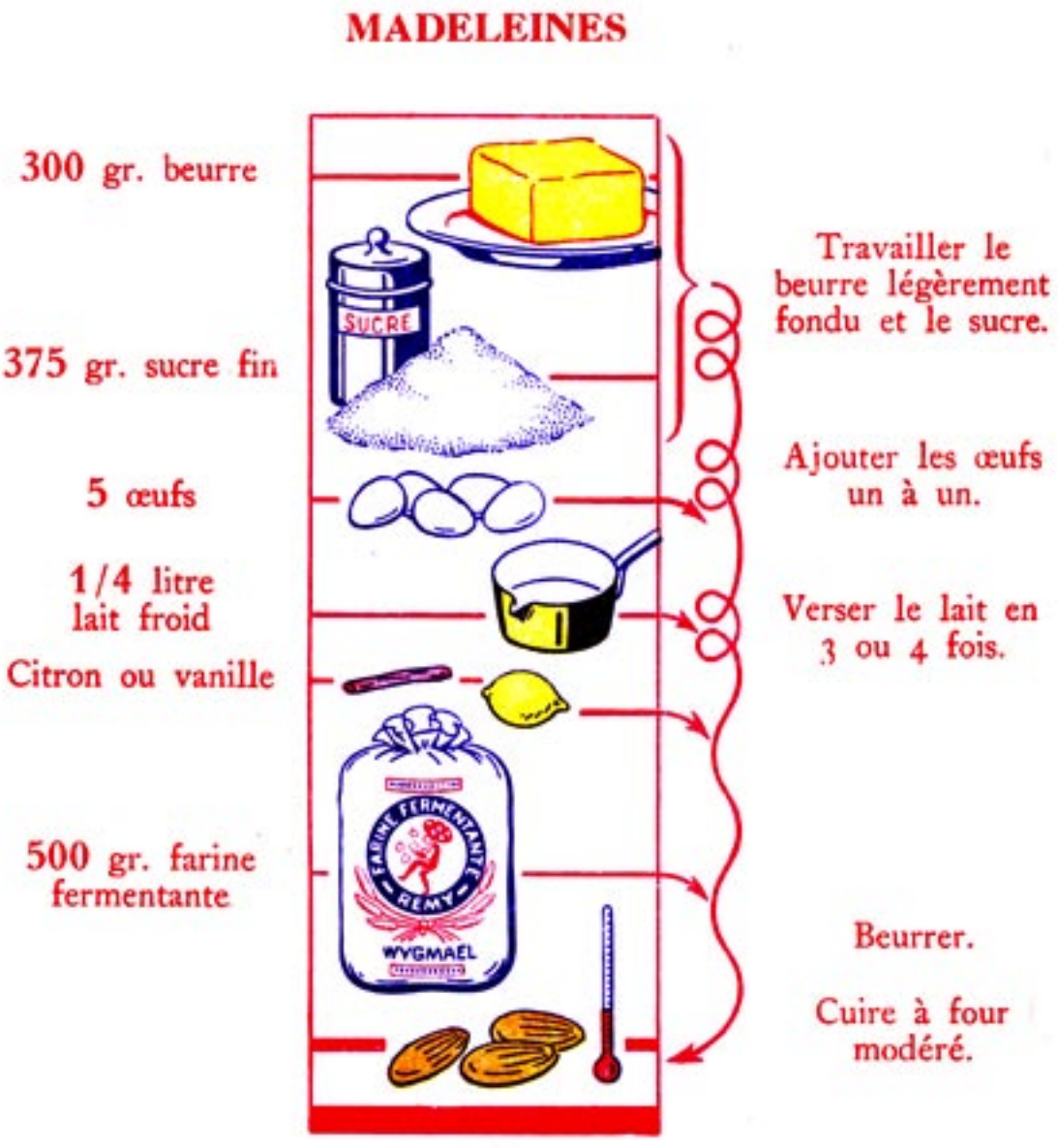
http://processing.org/

Hello
Hello world !

La **programmation** consiste à déterminer la démarche permettant d'obtenir, à l'aide d'un ordinateur, la solution à un problème donné.

Cette marche à suivre s'appelle un **algorithme**.

Une recette est un algorithme.



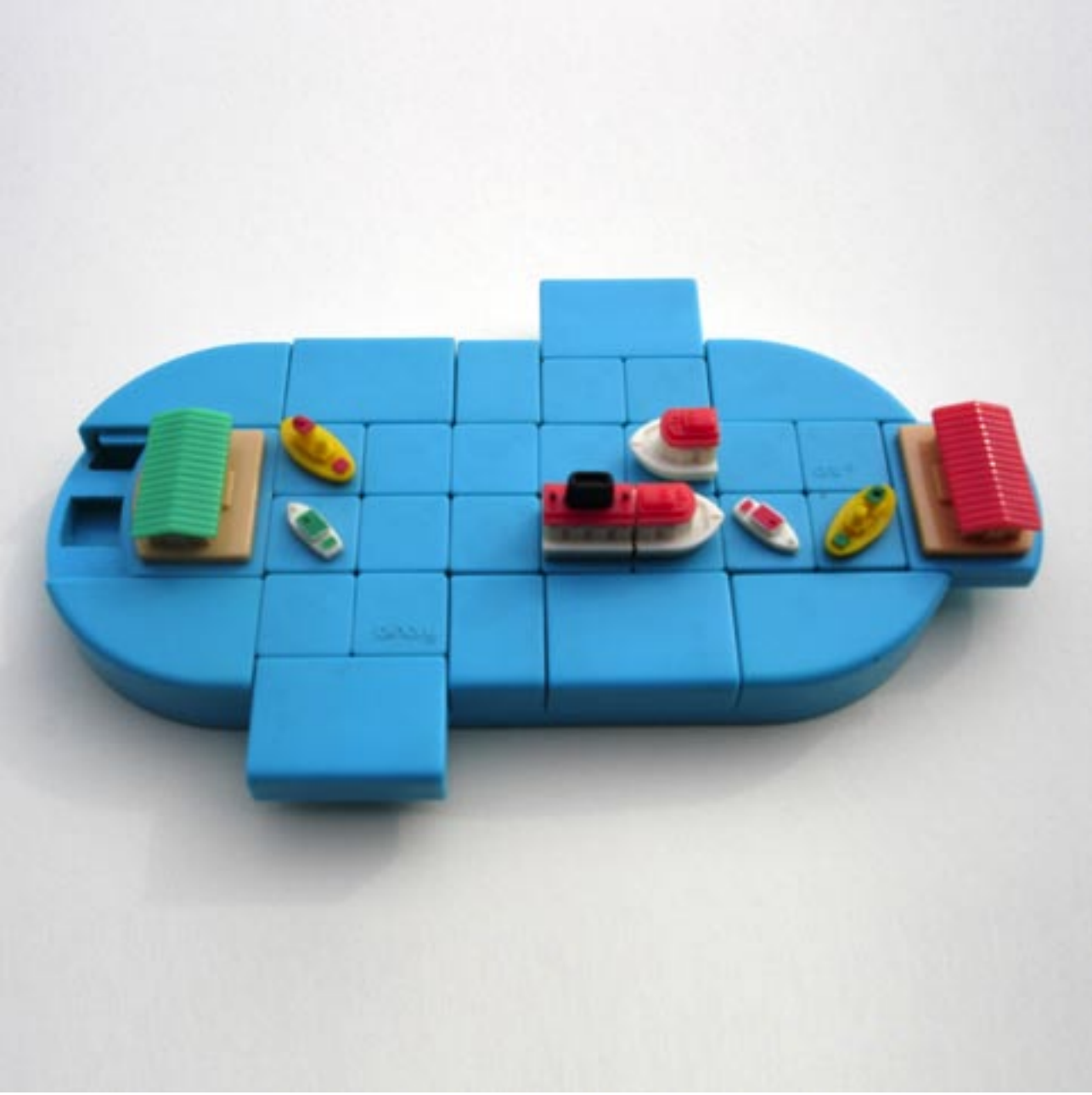
De **87 Rue du Page Bruxelles (Belgique)** A **10 Rue du Tabellion Bruxelles (Belgique)**

Itinéraire **Express** Distance **0.3 km** Voies express **0.0 km** Durée (1) **0H06**
Indemnités **0.0 €** Véhicule **Piéton**

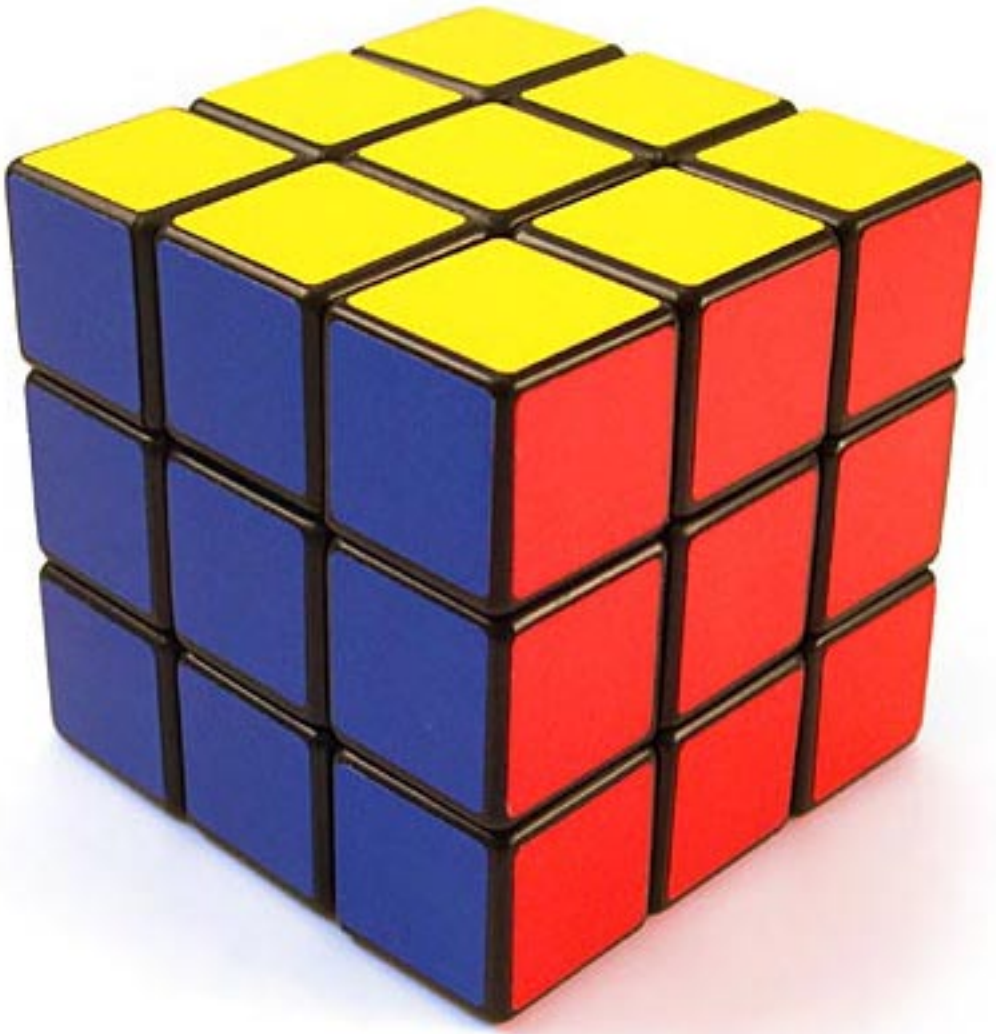
Cumul	Temps		Feuille de route
0 m	0H00		**87 Rue du Page Bruxelles (Belgique)**
			Continuer sur **Rue du Page [27m]**
27 m			Prendre à gauche **Rue Fourmois [120m]**
150 m	0H02		Prendre à droite **Rue du Tabellion [240m]**
390 m	0H06		**10 Rue du Tabellion Bruxelles (Belgique)**

Un mode d’emploi est un algorithme.

Fifteen Puzzle.



Port-to-Port.



Le Rubik’s cube, casse-tête inventé en 1974 par Ernö Rubik, sculpteur et architecte hongrois.
http://fr.wikipedia.org/wiki/Cube_de_Rubik

Le boulier chinois (suan pan): chaque tige correspond, de droite à gauche, respectivement aux unités, dizaines, centaines, milliers etc.
En 1945, Kiyoshi Matsuzaki, un comptable japonais muni d'un soroban (boulier japonais) remporta un match contre Nathan Woods, un opérateur de calculatrice électrique, par un score de 4 à 1.
http://en.wikipedia.org/wiki/Abacus

Man Ray et Marcel Duchamp disputant une partie d’échec.
http://www.marcelduchamp.net/

Gary Kasparov triomphe
contre le programme Deep Blue de IBM en février 1996.
Lors de la revanche, un an plus tard, il perd contre Deeper Blue.
http://www.research.ibm.com/deepblue/

Martin Wattenberg a réalisé un jeu d'échec avec PROCESSING :
http://www.turbulence.org/spotlight/thinking/chess.html

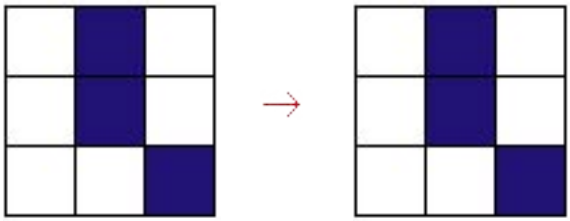
Le «Jeu de la vie» imaginé par John Horton Conway en 1970
est probablement le plus connu des automates cellulaires.
http://en.wikipedia.org/wiki/Conway's_Game_of_Life
http://www.metaphorical.net/code/processing/index2.html?page=life

Le jeu de la vie ne nécessite aucun joueur : il s'agit d'un automate cellulaire, c'est-à-dire un modèle où chaque état conduit mécaniquement à l'état suivant à partir de règles pré-établies.

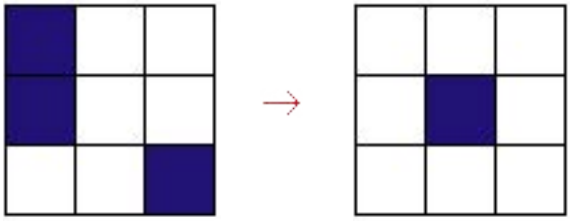
Le jeu se déroule sur une grille à deux dimensions, dont les cases (les «cellules») peuvent prendre deux états distincts : «vivantes» ou «mortes».

À chaque étape, l'évolution d'une cellule est entièrement déterminée par l'état de ses huit voisines de la façon suivante :

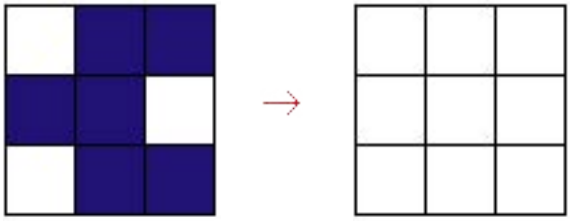
1- Si une cellule vivante (ici en bleu) est entourée de deux ou trois voisines vivantes, alors elle reste vivante. le reste, sinon elle meurt.



2- Si une cellule morte est entourée d'exactement trois voisines vivantes devient vivante (elle naît).



3- Dans tous les autres cas, la cellule meurt.



Pour plus de détails, lire l'article de Jean-Claude Heudin en annexe.

Voir un «Jeu de la vie» programmé avec PROCESSING par Michael Davis :
http://processing.org/learning/examples/cellularautomata1.html

Un autre programmé avec PROCESSING par William Ngan :
http://www.metaphorical.net/code/processing/index2.html?page=life